Amendments to the Claims:

1. (Original) A parallel pattern detection engine (PPDE) integrated circuit (IC) for detecting one or more patterns in a sequence of input data comprising:

an input/output (I/O) interface for coupling data into and out of the PPDE;

M processing units (PUs), each of the M PUs having compare circuitry for comparing each of the sequence of input data to a pattern stored in each of the M PUs and generating a compare output, wherein an address pointer selecting the pattern in each of the M PUs is modified in response to a logic state of the compare output and an operation code stored with the pattern;

an input bus for coupling the sequence of input data to each of the M PUs in parallel;

an output bus coupled to the I/O interface for sending output data to the I/O interface;

control circuitry coupled to the I/O interface and coupling control data on a control data bus and identification (ID) on an ID bus to each of the M processing units;

ID selection circuitry for selecting a match ID from ID data identifying the M PUs in response to a pattern match signal and match mode data, wherein the match ID and match data corresponding to the match ID are saved in a temporary register as the output data; and

cascade circuitry coupled from each of the M PUs to one or more adjacent PUs within the M PUs for selectively coupling chain data between one or more groups of two or more adjacent PUs selected from the M PUs in response to the control data.

2. (Original) The PPDE of claim 1 further comprising an input buffer coupled to the I/O interface for receiving and writing input data as parallel data at a write address.

3. (Original) The PPDE of claim 2 further comprising a multiplexer coupled to the input bus and the input buffer for sequentially coupling single data from the input buffer data to the input bus, wherein parallel data are selected using a read address.

4. (Original) The PPDE of claim 1 further comprising an output buffer coupled to the output bus and to the temporary register for receiving and writing output data to the output buffer at a write address and coupling output data to the output bus corresponding to a read address.

5. (Original) The PPDE of claim 1, wherein each of M processing units (PUs) has an ID register for storing a unique ID sent from the control circuitry.

6. (Original) The PPDE of claim 1, wherein each of M processing units (PUs) has a control register for storing the match mode data, wherein the match mode data determines criteria for generating the match signal and the match data.

7. (Original) The PPDE of claim 1, wherein each of the M PUs has a memory register array for storing a sequence of the pattern and corresponding operation codes addressed by an address register indexed by the address pointer.

8. (Original) The PPDE of claim 1, wherein the cascade circuitry enables the stored patterns of two or more PUs to be chained together as a single pattern using the chain data.

9. (Original) The PPDE of claim 9, wherein the chain data inhibits indexing the pointer of one PU until an adjacent PU coupled with the cascade circuitry has compared a last pattern to an input data.

10. (Original) The PPDE of claim 1, wherein the compare circuitry in each of the M PUs completes a compare of an input data to a selected pattern and generates a compare output and modifies the address pointer in the same cycle of a clock signal.

11. (Original) The PPDE of claim 1, wherein the match mode data for each of the M PUs sets a match mode comprising:

an exact match mode, wherein a pattern match indicates that the sequence of pattern matches a sequence of input data exactly,

a longest match mode wherein a pattern match indicates that a particular sequence of pattern corresponding to the match ID has the largest number of data in a sequence that compared to a sequence of data in the sequence of input data wherein the match data indicates the value of the largest number;

a maximum match mode, wherein a pattern match indicates that a particular sequence of pattern byes corresponding to the match ID has the largest number of data that compared in a broken sequence that compared to a broken sequence of input data, wherein the match data indicates the value of the largest number; and

a fuzzy match mode, wherein a pattern match indicates that a particular sequence of pattern corresponding to the match ID has the closet match to the sequence of input data as determined by a distance value, wherein the match data indicates the distance value.

12. (Original) The PPDE of claim 1, wherein the operation codes are selected from a set of operation codes comprising:

a match operation code indicating that the address pointer is incremented if the compare output is a logic one and the address pointer is reloaded to its initial value if the compare output is a logic zero;

an inverse operation code indicating that the address pointer is to be incremented if the compare output is a logic zero and the address pointer is reloaded to its initial value if the compare output is a logic one;

a wildcard operation code indicating that the address pointer is incremented if the compare output is a logic zero or a logic one;

a multiple wild card operation code indicating that the address pointer is to be held if the compare output is a logic zero otherwise the address pointer is incremented; and

a last operation code indicating that the address pointer is frozen until the matching process receives a reset if the compare output is a logic one and the address pointer is reloaded to its initial value if the compare output is a logic zero.

13. (Original) The PPDE of claim 1, wherein bits of the selected pattern are masked by a mask data stored in a mask register when the selected pattern is compared to an input data, the mask data indicating which bits of the selected pattern are not compared.

14. (Original) A method of determining if any of N sequences of pattern occurs within a sequence of input data using a parallel pattern detection engine (PPDE) comprising the steps of:

a) loading the N sequences of pattern into M processing units (PUs), each of the M PUs having compare circuitry for comparing each of the sequence of input data, in parallel, to a selected pattern in each of the N sequences of pattern stored in the M PUs;

b) loading identification (ID) data into each of the M PUs, wherein the ID data determines an ID for each of the M PUs;

c) loading match mode data into each of the M PUs setting criteria for determining when conditions have been met for indicating that one of the N sequences of pattern has been detected in the sequence of input data;

d) coupling a first input data in parallel to each of the M PUs;

e) comparing the first input data to the selected pattern determined by an address pointer in each of the M PUs and generating a compare output in each of the M PUs within a same clock cycle;

f) modifying the value of the address pointer in each of the M PUs in response to a logic state of the corresponding compare output and an operation code stored with the selected pattern in each of the M PUs;

g) selecting a match ID from the ID data in response to a pattern match signal indicating one of the N sequences of pattern has been detected;

h) storing the match ID and match data corresponding to the match ID; and

i) repeating steps (a-g) until a last input data of the sequence of input data has been compared.

15. (Original) The method of claim 14, wherein cascade circuitry is coupled from each of the M PUs to one or more adjacent PUs within the M PUs for selectively coupling chain data between two or more adjacent PUs selected from the M PUs in response to control data loaded into the two or more adjacent PUs.

16. (Original) The method of claim 14, one of the N sequences of pattern are partitioned and loaded into the two or more M PUs.

17. (Original) The method of claim 14, wherein each of M processing units (PUs) has a control register for storing the match mode data, wherein the match mode data determines criteria for generating the match signal and the match data.

18. (Original) The method of claim 17, wherein each of the M PUs has a memory register array for storing a sequence of the pattern and corresponding operation codes addressed by an address register indexed by the address pointer.

19. (Original) The method of claim 15, wherein the cascade circuitry enables the stored patterns of two or more PUs to be chained together as a single pattern using the chain data.

20. (Original) The method of claim 19, wherein the chain data inhibits indexing the pointer of one PU until an adjacent PU coupled with the cascade circuitry has compared a last pattern to an input data.

21. (Original) The method of claim 14, wherein compare circuitry in each of the M PUs completes a compare of an input data to a selected pattern and generates a compare output and modifies the address pointer in the same cycle of a clock signal.

22. (Original) The method of claim 14, wherein the match mode data for each of the M PUs sets a match mode comprising:

an exact match mode, wherein a pattern match indicates that the sequence of pattern matches a sequence of input data exactly;

a longest match mode wherein a pattern match indicates that a particular sequence of pattern corresponding to the match ID has the largest number of data in a sequence that compared to a sequence of data in the sequence of input data wherein the match data indicates the value of the largest number;

a maximum match mode, wherein a pattern match indicates that a particular sequence of pattern corresponding to the match ID has the largest number of data that compared to a broken sequence of input data, wherein the match data indicates the value of the largest number;

a fuzzy match mode, wherein a pattern match indicates that a particular sequence of pattern corresponding to the match ID has the closet match to the sequence of input data as determined by a distance value, wherein the match data indicates the distance value.

23. (Original) The method of claim 18, wherein the operation codes are selected from a set of operation codes comprising:

a match operation code indicating that the address pointer is incremented if the compare output is a logic one and the address pointer is reloaded to its initial value if the compare output is a logic zero;

an inverse operation code indicating that the address pointer is to be incremented if the compare output is a logic zero and the address pointer is reloaded to its initial value if the compare output is a logic one;

a wildcard operation code indicating that the address pointer is incremented if the compare output is a logic zero or a logic one;

a multiple wildcard operation code indicating that the address pointer is to be held if the compare output is a logic zero otherwise the address pointer is incremented; and

a last operation code indicating that the address pointer is frozen until the matching process receives a reset if the compare output is a logic one and the address pointer is reloaded to its initial value if the compare output is a logic zero.

24. (Original) A data processing system comprising:

a central processing unit (CPU);

a random access memory (RAM);

one or more parallel pattern detection engines (PPDEs); and

a bus coupling the CPU, RAM, and the one or more PPDEs, wherein each of the PPDEs has an input/output (I/O) interface for coupling data into and out of the PPDE;

M processing units (PUs), each of the M PUs having compare circuitry for comparing each of the sequence of input data to a pattern stored in each of the M PUs and generating a compare output, wherein an address pointer selecting the pattern data in each of the M PUs is modified in response to a logic state of the compare output and an operation code stored with the pattern data;

an input bus for coupling the sequence of input data to each of the M PUs in parallel;

an output bus coupled to the I/O interface for sending output data to the I/O interface;

control circuitry coupled to the I/O interface and coupling control data on a control data bus and identification (ID) on an ID bus to each of the M processing units;

ID selection circuitry for selecting a match ID from ID data identifying the M PUs in response to a pattern match signal and match mode data, wherein the match ID and match

data corresponding to the match ID are saved in a temporary register as the output data; and

cascade circuitry coupled from each of the M PUs to one or more adjacent PUs within the M PUs for selectively coupling chain data between one or more groups of two or more adjacent PUs selected from the M PUs in response to the control data.

25. (Original) The data processing system of claim 24, wherein each of M processing units (PUs) has an ID register for storing a unique ID sent from the control circuitry.

26. (Original) The data processing system of claim 24, wherein each of the M PUs has a memory register array for storing a sequence of the pattern and corresponding operation codes addressed by an address register indexed by the address pointer.

27. (Original) The data processing system of claim 24, wherein the cascade circuitry enables the stored patterns of two or more PUs to be chained together as a single pattern using the chain data.

28. (Original) The data processing system of claim 27, wherein the chain data inhibits indexing the pointer of one PU until an adjacent PU coupled with the cascade circuitry has compared a last pattern to an input data.

29. (Original) The data processing system of claim 24, wherein the compare circuitry in each of the M PUs completes a compare of an input data to a selected pattern and generates a compare output and modifies the address pointer in the same cycle of a clock signal.

30. (Original) The data processing system of claim 24, wherein the match mode data for each of the M PUs sets a match mode comprising:

an exact match mode, wherein a pattern match indicates that the sequence of pattern matches a sequence of input data exactly;

a longest match mode wherein a pattern match indicates that a particular sequence of pattern corresponding to the match ID has the largest number of data in a sequence that compared to a sequence of data in the sequence of input data wherein the match data indicates the value of the largest number;

a maximum match mode, wherein a pattern match indicates that a particular sequence of pattern corresponding to the match ID has the largest number of data that compared to a broken sequence of input data, wherein the match data indicates the value of the largest number; and

a fuzzy match mode, wherein a pattern match indicates that a particular sequence of pattern corresponding to the match ID has the closet match to the sequence of input data as determined by a distance value, wherein the match data indicates the distance value.

31. (Original) The data processing system of claim 30, wherein the operation codes are selected from a set of operation codes comprising:

a match operation code indicating that the address pointer is incremented if the compare output is a logic one and the address pointer is reloaded to its initial value if the compare output is a logic zero;

an inverse operation code indicating that the address pointer is to be incremented if the compare output is a logic zero and the address pointer is reloaded to its initial value if the compare output is a logic one;

a wildcard operation code indicating that the address pointer is incremented if the compare output is a logic zero or a logic one;

a multiple wildcard operation code indicating that the address pointer is to be held if the compare output is a logic zero otherwise the address pointer is incremented; and

a last operation code indicating that the address pointer is frozen until the matching process receives a reset if the compare output is a logic one and the address pointer is reloaded to its initial value if the compare output is a logic zero.

32. (Original) The data processing system of claim 30, wherein bits of the selected pattern are masked by a mask data stored in a mask register when the selected pattern is compared to an input data, the mask data indicating which bits of the selected pattern are not compared.

33. (New) A method to recognize objects comprising:
providing a database of patterns representative of objects to be recognized;
storing an operation code with predefined sector of said database;
generating a data stream representative of unknown objects;
providing an address pointer to select patterns in said database;
correlating a pattern selected with a block of data from said data stream; and
generating a status signal based upon result of the correlation and the operation code stored with the selected pattern.

34. (New) The method of claim 33 wherein correlation includes comparing the block of data from the data stream with the pattern selected from said database of patterns.

35. (New) The method of claim 33 wherein predefined sector includes eight bits.

36. (New) The method of claim 33 wherein a selected block of data includes eight bits.

37. (New) The method of claim 33 wherein said status signal, indicating unknown object identified, is being generated only if relationship between the data from said data stream and selected pattern is consistent with operation code stored with said

selected pattern.

38. (New) The method of claim 33 wherein said status signal, indicating unknown object not identified, is being generated only if relationship between the data from said data stream and the selected pattern is inconsistent with the operation code stored with said selected pattern.

39. (New) The method of claim 33 further including selectively modifying the address pointer in response to results of said correlation and the operation code stored with the selected pattern.

40. (New) The method of claim 33 wherein the objects include digitized representation of faces.

41. (New) The method of claim 33 or 34 wherein the block of data includes eight bits.

42. (New) A device comprising:
a memory;
a database of patterns stored in said memory;
a plurality of operation codes, with each one of said plurality of operation codes being associated with selected pattern stored in said memory;
a register for storing a sequence of input data;
a pointer to select patterns in said memory; and
a controller correlating data from said register with selected pattern from said database and issuing a status signal based upon result of the correlation and the operation code stored with said selected pattern.